



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um Algoritmo Direcionado a Bordas Que Utiliza Segmentação Para Interpolação de Imagens Digitais

Tiago Pigatto Lenza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia de Computação

Orientador

Prof. Dr. Bruno L. Macchiavello Espinoza

Brasília
2017

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Engenharia de Computação

Coordenador: Prof. Dr. Ricardo Pezzoul Jacobi

Banca examinadora composta por:

Prof. Dr. Bruno L. Macchiavello Espinoza (Orientador) — CIC/UnB

Prof. Dr. André Drummond — CIC/UnB

Prof. Dr. Flávio Vidal — CIC/UnB

CIP — Catalogação Internacional na Publicação

Lenza, Tiago Pigatto.

Um Algoritmo Direcionado a Bordas Que Utiliza Segmentação Para Interpolação de Imagens Digitais / Tiago Pigatto Lenza. Brasília : UnB, 2017.

103 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2017.

1. TEDI, 2. LAI, 3. TEDIs, 4. TEDia, 5. LATEDI

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um Algoritmo Direcionado a Bordas Que Utiliza Segmentação Para Interpolação de Imagens Digitais

Tiago Pigatto Lenza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia de Computação

Prof. Dr. Bruno L. Macchiavello Espinoza (Orientador)
CIC/UnB

Prof. Dr. André Drummond Prof. Dr. Flávio Vidal
CIC/UnB CIC/UnB

Prof. Dr. Ricardo Pezzoul Jacobi
Coordenador do Bacharelado em Engenharia de Computação

Brasília, 27 de janeiro de 2017

Dedicatória

Dedico este trabalho à minha família que sempre me apoiou durante todo o período em que eu estive cursando Engenharia de Computação e que sempre souberam me orientar e guiar sobre os meus afazeres. Dedico também as pessoas que sempre estiveram próximas à mim e que convivem comigo, sempre me consolando, orientando e guiando.

Agradecimentos

Agradeço ao professor Bruno por me orientar e me guiar neste processo de trabalho final, agradeço aos professores que passaram pela minha vida no meu período universitário que, de certa forma, também me orientaram academicamente e profissionalmente, e também agradeço à todas as pessoas as quais dediquei este trabalho.

Resumo

A área de interpolação de imagens digitais cresce constantemente e existem diversos trabalhos na literatura sobre o tema. Contudo, por melhor que sejam, esses sempre possuem alguma limitação ou podem ser melhorados. Neste trabalho, temos o foco de utilizar a segmentação para criar um novo método de interpolação, visando melhoramentos nas métricas numéricas, na parte visual e no desempenho, sempre tentando preservar as bordas da imagem original. São utilizados como base três algoritmos que tem o foco na interpolação de imagens com preservação de bordas: o *Locally Adaptive Non-Linear Interpolation* (LAI), o *True Edge-Directed Interpolation* (TEDI) e o *Locally Adaptive True Edge-Directed Interpolation* (LATEDI), que mistura um pouco dos conceitos dos primeiros. Iremos acrescentar ao TEDI uma segmentação inicial, dando origem ao $TEDI_s$ e iremos mesclar o conceito de segmentação com uma mistura do LAI e do TEDI, resultando no $TEDI_a$.

Palavras-chave: TEDI, LAI, TEDIs, TEDIa, LATEDI

Abstract

The digital image interpolation field of study grows constantly and there are several works on the literature about it. However, as good as this studies may be, they always have some limitations or lack of improvement. In this work, we focus on using segmentation to create a new interpolation method, aiming to improve the metrics, the visual comfort and the performance, but always trying to maintain the edges from the original image. We use three algorithms that have focus on edge preserving: Locally Adaptive Non-Linear Interpolation (LAI), True Edge-Directed Interpolation (TEDI) and Locally Adaptive True Edge-Directed Interpolation (LATEDI), which mixes the concepts of the first ones. We will add to TEDI an initial segmentation which will lead to the algorithm $TEDI_s$ and we will mix the concept of segmentation with the combination of LAI and TEDI, resulting on the algorithm $TEDI_a$.

Keywords: TEDI, LAI, TEDIs, TEDIa, LATEDI

Sumário

1	Introdução	1
1.1	Imagens médicas	1
1.2	Processamento de imagens médicas	2
1.3	Objetivos	4
2	Fundamentação teórica	5
2.1	Interpolação de imagens digitais	5
2.2	Segmentação de imagens	11
3	Imagem por ressonância magnética	14
3.1	Imagens abdominais	15
4	Revisão da literatura	17
4.1	<i>True Edge-Directed Interpolation</i> (TEDI)	18
4.2	<i>Locally Adaptive Non-Linear Interpolation</i> (LAI)	21
4.3	<i>Locally Adaptive True Edge-Directed Interpolation</i> (LATEDI)	24
5	Metodologia	26
5.1	$TEDI_s$	27
5.2	$TEDI_a$	28
6	Resultados	32
7	Conclusões	39
7.1	Trabalhos futuros	39
	Referências	40

Lista de Figuras

1.1	Ferramenta de auxílio de diagnóstico.[7]	2
2.1	Comparação entre interpolação linear e polinomial	5
2.2	Imagem Original	6
2.3	Imagem aumentada antes da interpolação	6
2.4	Imagem interpolada pelo método Bilinear	6
2.5	Imagem interpolada pelo método <i>Nearest Neighbor</i>	6
2.6	Imagem original	8
2.7	Imagem aumentada	8
2.8	<i>Nearest Neighbor</i>	8
2.9	Valor P a ser interpolado[5]	9
2.10	Imagem original	10
2.11	Interpolação bilinear	10
2.12	Interpolação bicúbica	10
2.13	<i>Nearest neighbor</i>	11
2.14	<i>Bilinear interpolation</i>	11
2.15	<i>Bicubic interpolation</i>	11
2.16	Exemplo do algoritmo <i>k-means</i> em cima de um conjunto de dados aleatório	12
3.1	Exemplos de imagens por ressonância magnética[20]	15
3.2	Exemplos de imagens do abdômen por ressonância magnética[19]	16
4.1	Imagem original	19
4.2	Mapa de bordas	19
4.3	Bordas 0°- 45°(RH)	20
4.4	Bordas 45°- 90°(LV)	20
4.5	Bordas 90°- 135°(RV)	20
4.6	Bordas 135°- 180°(LH)	20
4.7	Máscaras aplicadas sobre os pontos analisados nas Figuras 4.3 4.4 4.5 4.6	21
4.8	Fase de expansão do LAI	22
4.9	Nomenclatura dos pontos da imagem expandida	22
4.10	Vizinhança dos pontos com ao menos uma coordenada ímpar	23
4.11	Fase de expansão do LATEDI	25
5.1	Imagem original	27
5.2	Imagem segmentada	27
5.3	Imagem original	27
5.4	Imagem segmentada	27

5.5	Primeira etapa do $TEDI_s$	28
5.6	Primeira etapa do $TEDI_s$	29
5.7	Vizinhança 5x5 considerada	30
6.1	Imagem original	33
6.2	TEDI	34
6.3	LAI	34
6.4	<i>New Edge-Directed Interpolation</i> (NEDI)	34
6.5	<i>Edge-Guided Image Interpolation</i> (EGII)	34
6.6	$TEDI_a$	34
6.7	$TEDI_s$	34
6.8	LAI	35
6.9	TEDI	35
6.10	$TEDI_a$	35
6.11	$TEDI_s$	35
6.12	Comparação entre as bordas das imagens geradas pelos respectivos algoritmos	36
6.13	<i>Peak Signal-to-Noise Ratio</i> (PSNR)	36
6.14	<i>Structural Similarity Index Measure</i> (SSIM)	37
6.15	<i>Feature Similarity Index Measure</i> (FSIM)	37

Lista de Tabelas

6.1	Média dos valores obtidos com cada algoritmo	38
6.2	Tempo de execução médio dos algoritmos	38

Lista de Abreviaturas e Siglas

EGII *Edge-Guided Image Interpolation*. vii, 18, 32, 34, 38

FSIM *Feature Similarity Index Measure*. vii, 32, 37, 38

ICBI *Iterative Curve Based Interpolation*. 18

LAI *Locally Adaptive Non-Linear Interpolation*. v–vii, 21, 22, 24, 25, 30, 32, 34, 35, 37, 38

LATEDI *Locally Adaptive True Edge-Directed Interpolation*. v, vi, 22–25

NEDI *New Edge-Directed Interpolation*. vii, 17, 32, 34, 38

PSNR *Peak Signal-to-Noise Ratio*. vii, 3, 32, 36, 38

SSIM *Structural Similarity Index Measure*. vii, 32, 37, 38

TEDI *True Edge-Directed Interpolation*. v, vii, 3, 4, 18, 19, 21, 24, 25, 32, 34, 35, 37–39

Capítulo 1

Introdução

1.1 Imagens médicas

Com o rápido avanço tecnológico da medicina, pode-se observar a grande quantidade de técnicas e algoritmos computacionais que são desenvolvidos exclusivamente para o uso médico. Softwares para gestão de pacientes, utilização de inteligência artificial na tomada de decisões médicas [25], robôs para ajudar em cirurgias delicadas [3], entre outras diversas áreas que vêm se desenvolvendo com o foco em realizar um atendimento médico com maior qualidade.

A captura de imagens médicas tem propósitos tanto para diagnóstico como para tratamento. Essas imagens podem ser capturadas por diferentes tecnologias como raios-x, ultrassom, ressonância magnética, fármacos radioativos (utilizados na medicina nuclear) ou métodos que utilizam a luz (endoscopia). Estes métodos permitem o acompanhamento médico de um paciente para que possa ser realizado o diagnóstico e a avaliação do tratamento.

A tecnologia por trás destes sensores e técnicas vêm crescendo muito rápido desde que o raio-x foi inventado em 1895 [6]. Hoje temos tecnologia onde o paciente fica exposto aos raios-x por questão de milissegundos, reduzindo drasticamente a exposição do paciente. A qualidade dessas imagens também melhorou, o que facilita o diagnóstico e fornece maior quantidade de detalhes. Porém, a tecnologia de raios-x possui uma limitação de detalhes, o que não possibilita a análise de veias e órgãos, impossibilitando assim alguns diagnósticos.

As imagens digitais trouxeram um grande avanço para a medicina, possibilitando a visualização em tempo real de imagens raio-x, o que ajudava na observação de procedimentos em tempo real. Este tipo de imagem também trouxe técnicas como a tomografia e a ressonância magnética, que pode ser utilizada para observar a atividade do cérebro humano e analisar traumas sofridos.

A cada ano as técnicas de imagens médicas avançam cada vez mais. Hoje temos técnicas como a *Optical coherence tomography* (OCT), que é uma espécie de tomografia que constrói a imagem pela luz transmitida através do corpo. Também utiliza-se ultrassom e tomografia com a ajuda de micro-bolhas, que ajudam a dar um maior contraste nas imagens e observar áreas específicas do corpo humano.

1.2 Processamento de imagens médicas

Neste âmbito tecnológico, uma área que vem se destacando muito, tanto pelo seu crescimento rápido como pelo seu potencial inovador, é a de processamento de imagens. Imagens médicas fazem parte de uma grande realidade no campo da medicina moderna. Imagens internas do corpo humano como de órgãos e ossos ajudam no diagnóstico de diversas doenças que não são facilmente identificadas. As imagens médicas geralmente vêm acompanhadas de muito ruído, o que dificulta sua visualização e também a conclusão sobre a imagem. Para melhorar a qualidade delas, utiliza-se o processamento de imagens. O campo de processamento de imagens digitais se refere ao processamento de imagens digitais com a utilização de um computador digital. Uma imagem pode ser definida como uma função bidimensional $f(x,y)$, onde x e y são coordenadas espaciais e a amplitude de f em qualquer par de coordenada (x,y) é chamada de intensidade ou nível de cinza da imagem naquele ponto. Quando x , y e os valores de intensidade de f forem finitos (quantidades discretas), chamamos a imagem de imagem digital. [10]

O processamento de imagens médicas inclui a melhoria e disponibilização de imagens adquiridas através de raio-x, ultrassom, ressonância magnética, imagens de medicina nuclear, ou qualquer outra tecnologia de obtenção de imagem. Também é possível a reconstrução de imagens e a criação de imagens 3D. Originalmente, quando a tomografia foi inventada, demorava-se 24 horas para reconstruir uma imagem a partir dos dados obtidos, atualmente, graças ao avanço das técnicas de processamento de imagem e também do poder computacional das máquinas, leva-se segundos [17].

O processamento de imagens também ajuda na formulação de diagnósticos. Em vez de uma observação médica, a imagem pode ser analisada por um computador e ter um diagnóstico no momento em que foi produzida. Estas técnicas também possibilitam descobrir o diâmetro, tamanho, vascularidade, entre outros parâmetros de órgãos ou até de tumores.

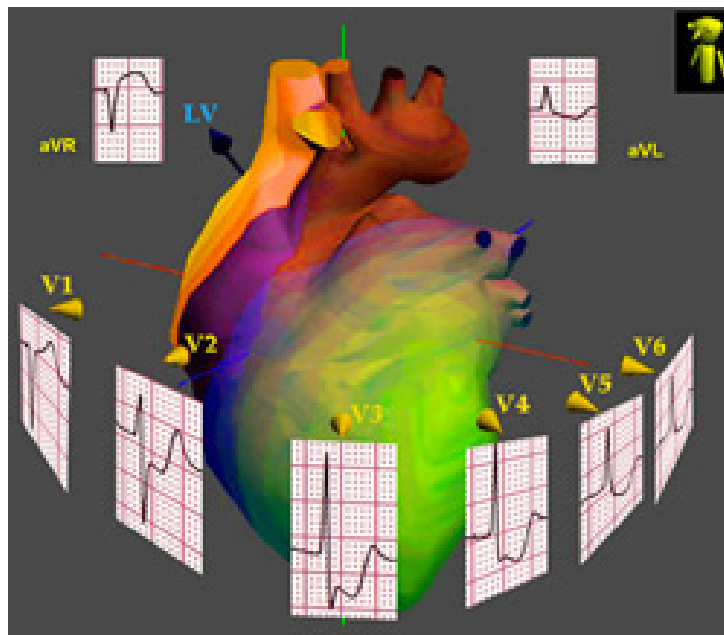


Figura 1.1: Ferramenta de auxílio de diagnóstico.[7]

Na área médica, as imagens produzidas podem vir muito distorcidas, assim como com qualidade muito baixa e, muitas vezes, o tempo para que esta imagem seja produzida é crítico. Outro grande problema pode ser o armazenamento destas imagens, pois em geral elas possuem um tamanho muito superior às imagens comuns. O processamento de imagens se encaixa em várias situações como compressão das imagens médicas para armazenamento, melhoramento da qualidade da imagem para que o diagnóstico seja mais eficiente, zoom em imagens médicas com menor perda de qualidade, entre várias outras aplicações.

Uma área muito explorada é a interpolação de imagens, que consiste em estimar o valor de um pixel que não existe baseado em pixels com valores conhecidos. A interpolação é muito utilizada quando se quer dar zoom em imagens ou quando se quer modificar o tamanho de alguma imagem. Algumas técnicas de interpolação de imagens são: vizinho mais próximo, interpolação bilinear e interpolação bicúbica.

O vizinho mais próximo consiste em substituir o valor do pixel em questão pelo valor do pixel que está mais próximo dele. A técnica de interpolação bilinear consiste em aplicar a interpolação linear em dois sentidos diferentes, pois a imagem possui duas dimensões. A interpolação bicúbica consiste em somar os valores ao redor do pixel em questão utilizando uma aproximação cúbica. A interpolação bicúbica gera um efeito muito mais suave na imagem do que os outros.

A técnica de interpolação de imagens é explorada por diversos autores gerando diversas aplicações interessantes para o uso desta. A interpolação de imagens focada em bordas através de filtros direcionais e fusão de dados [27], interpolação com convolução cúbica [12], interpolação de imagens por modelo auto-regressivo 2D [29], entre outros.

A qualidade de bordas em imagens médicas é crucial para alguns diagnósticos pois, muitas vezes, é necessário a ampliação da imagem, e isto pode causar uma deformação nas bordas. A qualidade da imagem obtida pode não ser satisfatória e possuir bordas mal-formadas, o que pode levar a um diagnóstico equivocado, pois as bordas fornecerem detalhes importantes às imagens e ajudam a segmentar regiões visualmente diferentes. A visualização de bordas pode ajudar em detecção de formas e formatos de diversas partes do corpo. Uma técnica recente que utiliza interpolação para tentar restaurar as bordas de uma imagem é o **TEDI** (*true edge-directed interpolation*)[26]. Este método foca justamente em melhorar as bordas da imagem para que a visualização desta seja melhor. O foco desta técnica é a proporção entre qualidade e desempenho, visto que em alguns casos a velocidade com que a imagem é processada pode ser crucial. Como por exemplo em situações de risco de vida ou em acidentes traumáticas, onde a velocidade em identificar o problema é essencial.

O objetivo é mudar o modo como o **TEDI** foi abordado e tornar o algoritmo muito mais eficaz no quesito de qualidade de imagem. Para medir esta diferença na melhora da imagem, iremos utilizar diversas métricas, entre elas a **PSNR**. A **PSNR** é a relação sinal-ruído em uma imagem. Se a quantidade de ruído na imagem for muito alto para a quantidade de dados que esta possui, sua **PSNR** será baixa, ou seja, a qualidade da imagem não será satisfatória. Caso a quantidade de ruído seja baixo em comparação à quantidade de sinal obtido(a imagem em si), a **PSNR** será alta. O objetivo é obter métricas de qualidade superiores ao **TEDI**.

1.3 Objetivos

Visto a grande necessidade de conciliar o desempenho e a qualidade, o trabalho tem por finalidade melhorar a visibilidade das bordas da imagem original. Para isso serão utilizadas tanto imagens de baixa e alta qualidade para aumentá-las em uma escala significativa (que seja visualmente útil à quem sirva), sem que ocorra uma perda significativa da qualidade e com uma preservação das bordas.

Serão apresentadas duas vertentes do algoritmo **TEDI**. As duas versões serão modificadas para possuírem uma segmentação inicial das imagens. A primeira versão, $TEDI_S$, será muito parecida com o **TEDI**, apenas com o conceito de segmentação adicionado. A segunda versão, $TEDI_A$, terá uma modificação nos filtros que são aplicados para a interpolação das bordas.

A demonstração dos resultados se dará pela utilização desta técnica em um conjunto de imagens médicas e comparando os resultados com resultados posteriormente obtidos por outros autores. Serão utilizadas métricas comuns ao processamento de imagens para obter um resultado numérico e que possa ser facilmente comparado.

Para alcançar o resultado desejado foram determinados dois objetivos específicos:

- Obter métricas superiores ao **TEDI** e aos algoritmos que serão utilizados para comparação;
- Obter uma qualidade visual mais agradável em relação às bordas das imagens.

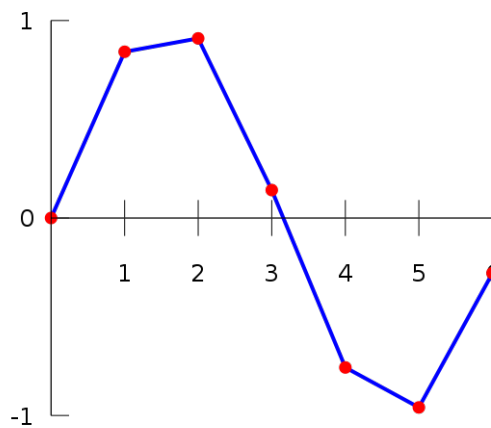
Para se atingir os objetivos, será proposto um conceito modificado do algoritmo **TEDI**.

Capítulo 2

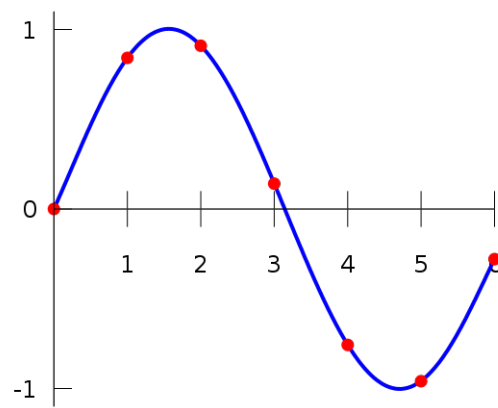
Fundamentação teórica

2.1 Interpolação de imagens digitais

A técnica de interpolação, em matemática, refere-se ao ato de construir ou manipular um conjunto de dados a partir de um conjunto de dados discretos já conhecidos. Assim, pode-se estipular um dado desconhecido a partir do conjunto conhecido. Essas informações, quando utilizadas em alguma área prática, são obtidas através de algum tipo de amostragem resultante de um experimento. Os dados obtidos neste experimento, podem não ser contínuos e podem dificultar a modelagem desses. A interpolação pode agrupar estas informações em funções que se aproximem aos dados obtidos. A interpolação também permite a aproximação de funções mais complexas para funções mais simples e mais fáceis de serem manipuladas. A interpolação em si geralmente não é o objetivo final, e sim um passo para alcançá-lo.



(a) Interpolação linear.



(b) Interpolação polinomial grau 1.

Figura 2.1: Comparação entre interpolação linear e polinomial

As três hipóteses mais importantes para que se possa usar interpolação são[21]:

1. Os dados utilizados são definidos como contínuos;
2. É possível calcular o valor da função contínua dado qualquer parâmetro de entrada;

3. A avaliação da função contínua nos pontos de amostragem produz o mesmo valor que os próprios dados.

Em imagens digitais, várias operações podem ser realizadas para mudar o modo como estas são visualizadas, por exemplo, aumentar, rotacionar ou diminuir. Essas operações resultam em uma imagem completamente diferente e, se analisarmos os pixels desta nova imagem, eles serão completamente diferentes dos pixels da anterior. Para isso, pode-se usar uma técnica de interpolação para que as imagens fiquem mais suaves. Vale lembrar que, mesmo após a suavização ou qualquer outro tipo de operação, a qualidade nunca será a mesma da imagem original, pois na prática é impossível uma recuperação perfeita.

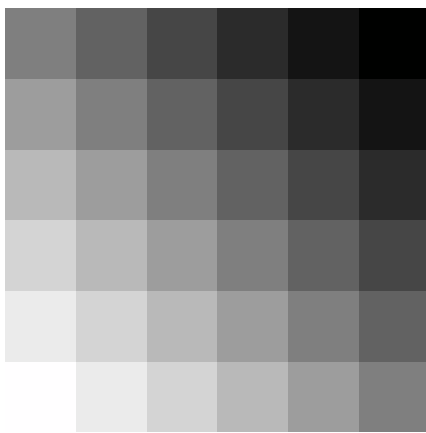


Figura 2.2: Imagem Original

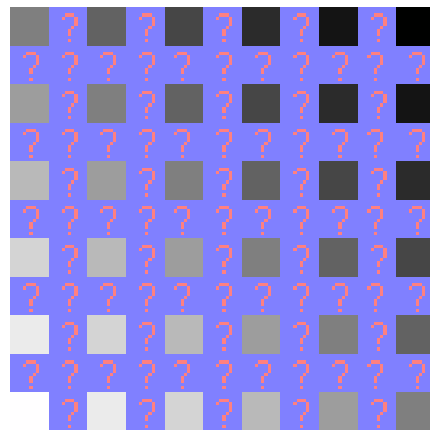


Figura 2.3: Imagem aumentada antes da interpolação

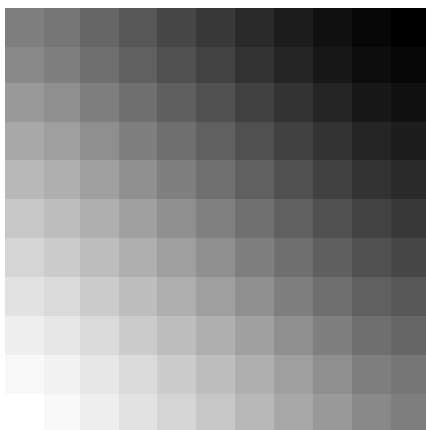


Figura 2.4: Imagem interpolada pelo método Bilinear

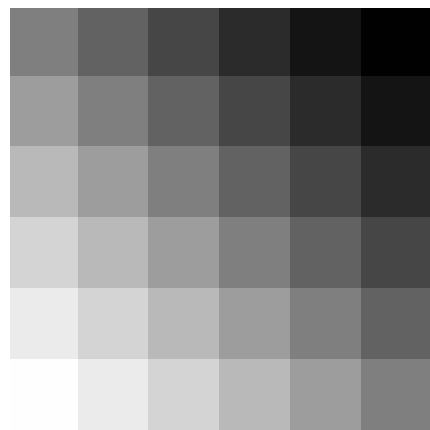


Figura 2.5: Imagem interpolada pelo método *Nearest Neighbor* [11]

É fácil notar que a Figura 2.4 possui uma mudança muito mais suave do que a Figura 2.5, sendo que ambas foram aumentadas na mesma proporção.

Existem diversas aplicações onde a busca por uma qualidade de imagem não seria possível de se obter sem o uso da interpolação. Um exemplo, é a área biomédica, onde muitas vezes precisa-se ampliar uma imagem ou modificar a taxa de pixels. Isto pode ser utilizado em um *scanner* que não possui uma resolução homogênea para corrigir a resolução na imagem final.

Outro método onde a interpolação é muito utilizada acontece no *reslicing*[14] que consiste em dividir várias vezes uma imagem com mais de uma dimensão em diferentes imagens. Esta técnica pode ser empregada em imagens obtidas por ressonância magnética.

Existe uma vasta área de aplicações para interpolação diferentes das citadas acima como renderização de volume[24], multiresolução[22] e transformações geométricas genéricas.

Existem diversos algoritmos de interpolação e cada um produz um efeito diferente e adequado para um tipo de aplicação. Os mais utilizados são:

- *Nearest neighbor interpolation*
- *Bilinear interpolation*
- *Bicubic interpolation*
- *B-Spline interpolation*
- *Sinc interpolation*

Os métodos *B-Spline* e *Sinc*, são métodos que abrangem uma retenção maior de dados da imagem, levando à um gasto computacional muito maior, enquanto os outros são métodos mais simples porém muito mais rápidos de serem executados, podendo ser combinados com diversas técnicas.

O *Nearest neighbor interpolation* é um método básico porém muito útil em aplicações simples, pois é o método que requer o menor tempo de processamento. Este método consiste em basicamente atribuir um valor ao pixel em questão pelo valor do seu vizinho mais próximo.

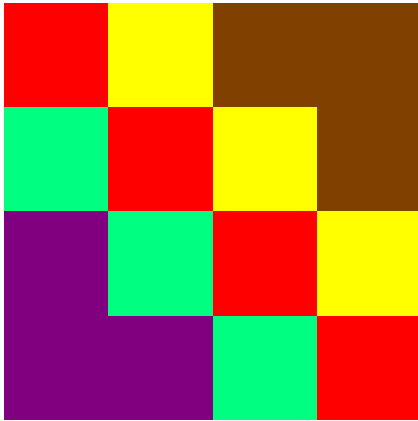


Figura 2.6: Imagem original

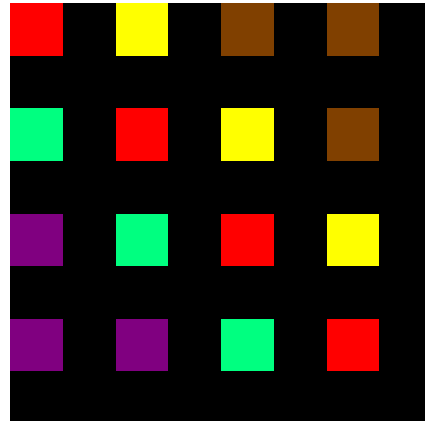
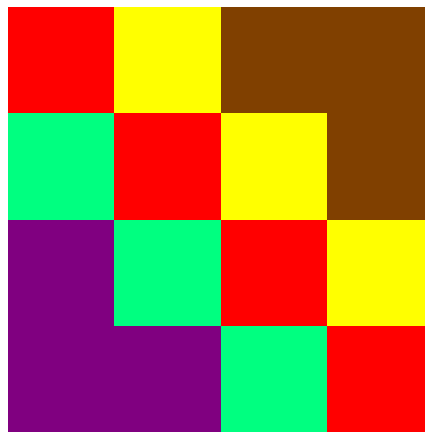


Figura 2.7: Imagem aumentada



[1]

Figura 2.8: *Nearest Neighbor*

Podemos observar o funcionamento do algoritmo na Figura 2.7. Os pixels que estão em preto foram preenchidos pelos seus vizinhos mais próximos resultando na Figura 2.8.

A interpolação bilinear pode ser usada para estimar o valor de um pixel desconhecido após uma operação geométrica em uma imagem. Ela também é uma técnica básica de *re-sampling* em processamento de imagens. Outra utilização deste método é o filtro bilinear, que é utilizado em mapeamento de texturas para produzir uma imagem mais realística após a operação. Essa técnica é um pouco mais complexa que o método *nearest neighbor*, porém resulta em uma imagem com a aparência muito mais suave. A interpolação bilinear considera uma vizinhança 2x2 em volta do pixel de valor desconhecido. A partir desta vizinhança, atribui-se um peso a cada posição e calcula-se a média.

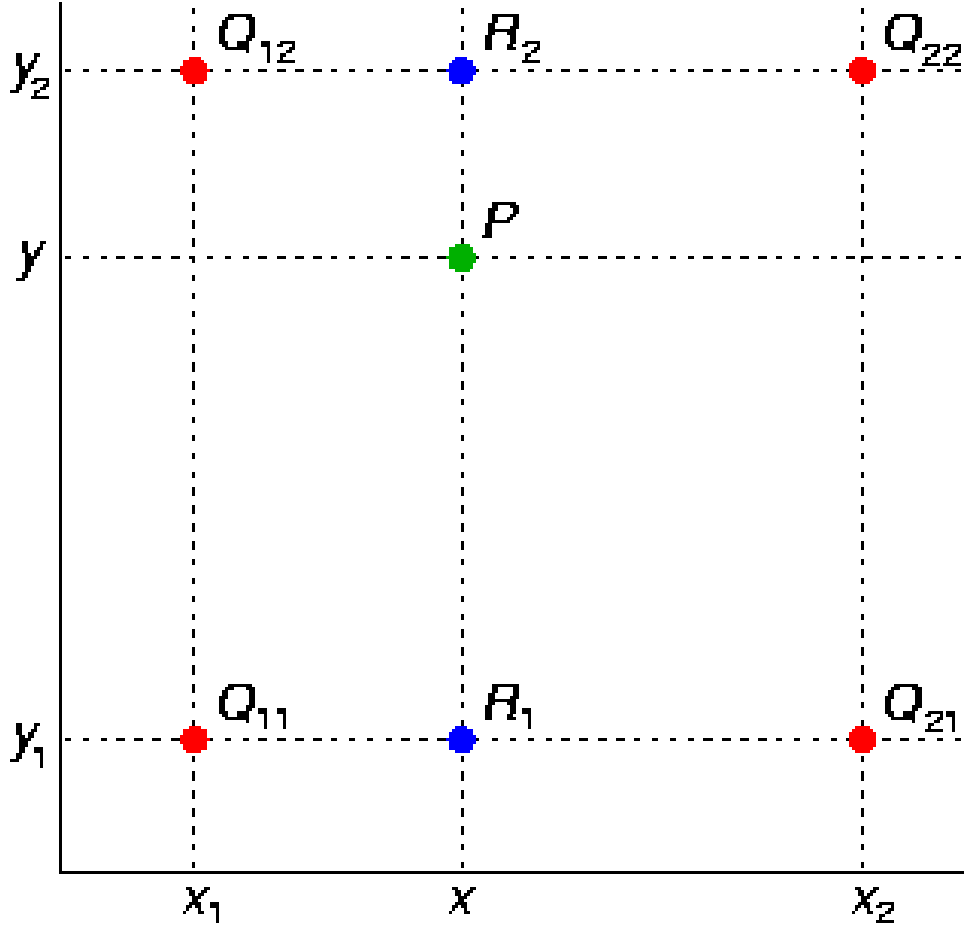


Figura 2.9: Valor P a ser interpolado[5]

Observando a Figura 2.9 temos o valor P que é o pixel a ser interpolado e os valores Q11, Q12, Q21 e Q22, que é a vizinhança 2x2 considerada. Os valores R1 e R2 são a média ponderada de Q11, Q21 e Q12, Q22, respectivamente. Os valores y , y_1 , y_2 , x , x_1 , x_2 representam valores cartesianos. Assim, a fórmula final para se calcular a interpolação bilinear destes pixels é:

$$P = \frac{y_2 - y}{y_2 - y_1} * R_1 + \frac{y - y_1}{y_2 - y_1} * R_2 \quad (2.1)$$

O resultado é muito parecido com a média de todos os valores dos vizinhos, porém vale lembrar que, se algum dos vizinhos estiver muito afastado, o seu peso será menor.

Um passo além da interpolação bilinear está a interpolação bicúbica. Ela considera uma vizinhança 4x4 em vez de 2x2, o que resulta em 16 pixels de vizinhança ao vez de 4. Isto gera uma imagem com bordas muito mais bem definidas e com previsões do pixel em questão mais corretas. Em termos de processamento o algoritmo com certeza seria mais custoso do que a interpolação bilinear, por exemplo. Porém, se analisarmos o custo por benefício da interpolação bicúbica, pode-se observar um ganho muito grande em qualidade, por isso, vários programas de edição de imagem atualmente utilizam este método.

A fórmula para se calcular a interpolação bicúbica é:

$$p(x, y) = \sum_{i=0}^n \sum_{j=0}^n a_{ij} x^i y^j \quad (2.2)$$

Onde i e j representam as coordenadas da matriz 4x4, a representa a intensidade do ponto, x e y representam as coordenadas do ponto na imagem.

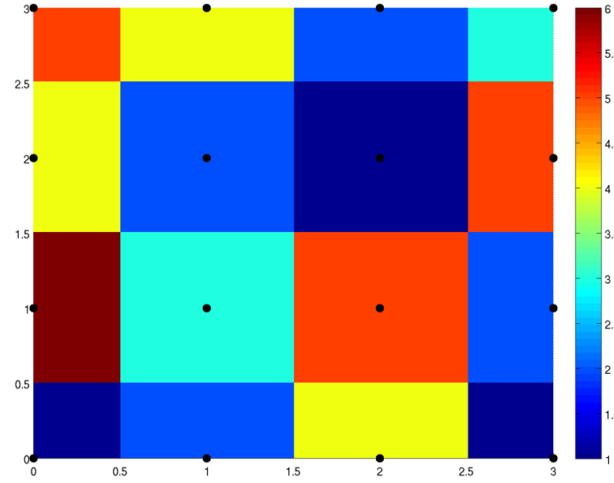


Figura 2.10: Imagem original

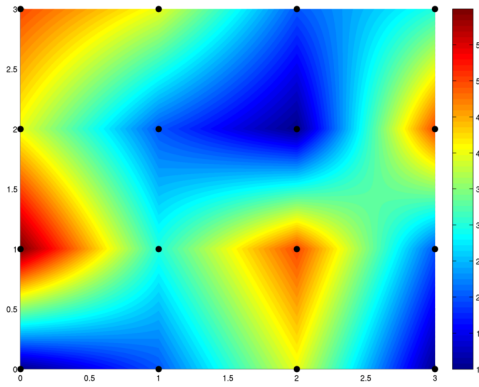


Figura 2.11: Interpolação bilinear

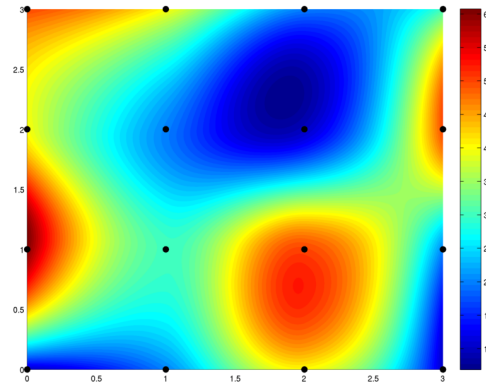


Figura 2.12: Interpolação bicúbica

Observados as Figuras 2.11 e 2.12 podemos observar a diferença visual entre a interpolação bilinear e a bicúbica. A segunda possui uma fluidez maior na troca de cor do que a primeira, o que acaba deixando a imagem mais suave. Outra coisa que pode-se observar é a definição da borda. Na imagem que foi produzida utilizando a interpolação bicúbica as margens estão claramente mais bem definidas.

Destes três métodos apresentados, a interpolação bicúbica com certeza é a mais utilizada, por oferecer um meio termo entre desempenho e qualidade: o desempenho é praticamente semelhante ao dos outros em quanto a qualidade é bem maior. Em algumas aplicações, a interpolação bilinear pode ser suficiente caso a necessidade por qualidade não seja grande e o desempenho seja crucial.

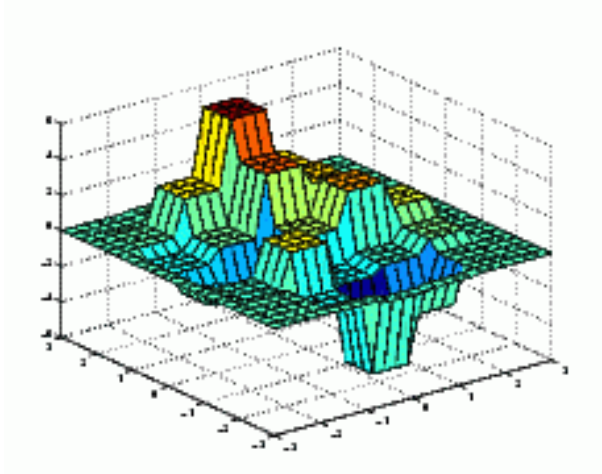


Figura 2.13: *Nearest neighbor*

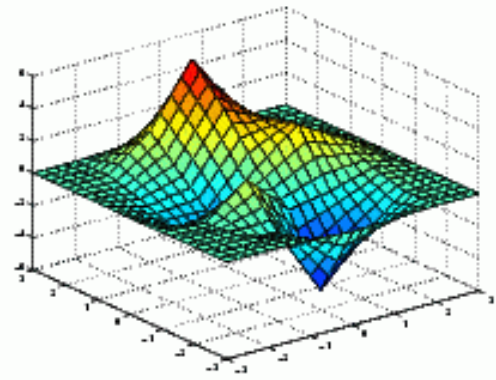


Figura 2.14: *Bilinear interpolation*

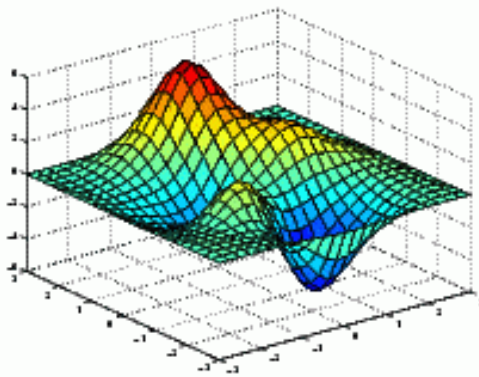


Figura 2.15: *Bicubic interpolation*

Analisando as Figuras 2.13, 2.14 e 2.15 podemos observar a real melhora em suavidade da interpolação bicúbica. O *nearest neighbor* produz um resultado com quase nenhuma suavidade, onde os valores pulam de um para o outro. A interpolação bilinear também apresenta uma certa suavidade, porém pode-se notar alguns picos na imagem, o que representam mudanças bruscas de valores.

2.2 Segmentação de imagens

O processo de segmentação, consiste basicamente em separar uma imagem digital em vários segmentos não sobrepostos [18]. A segmentação pode ser usada para a identificação de objetos, separação de regiões por bordas ou agrupamento de pixels. Ela é tipicamente utilizada para localizar objetos e limites (linhas, bordas, curvas) em imagens. Basicamente, este processo separa e nomeia a imagem em regiões. Uma mesma região compartilha de informações parecidas como cor, textura, intensidade. Regiões diferentes costumam possuir características bem distintas. No âmbito de imagens médicas, pode-se usar segmentação para reconhecimento de órgãos e tumores ou recriação de modelos 3D.

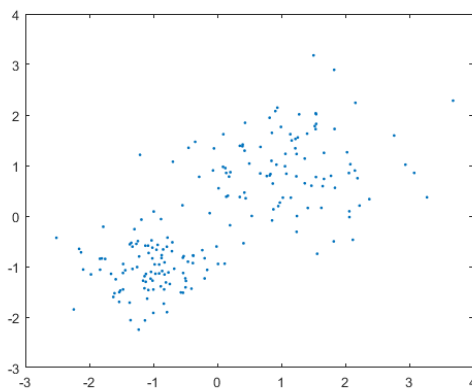
Neste trabalho, testamos alguns métodos que nos deram uma resposta melhor aos outros e por isso iremos tratar deles:

- *Thresholding*
- Clusterização
- Histograma

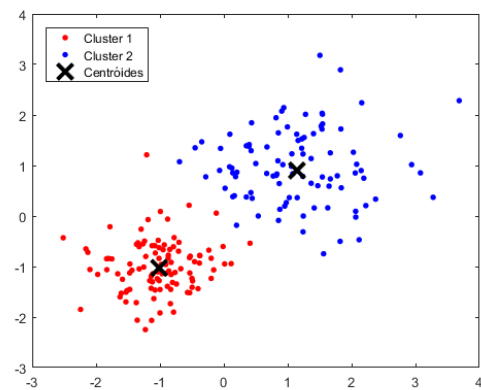
O método de *Thresholding* é o mais simples dos citados acima e pode ser executado de duas maneiras diferentes: com apenas um valor de limiar e com múltiplos valores de limiar. Este método consiste em estabelecer valores-limite para que a imagem seja dividida em partes. Na primeira alternativa, a imagem é binarizada, ou seja, valores acima de um limiar K irão ficar preto e valores abaixo deste limiar irão ficar branco, ou vice-versa. Na segunda abordagem, definem-se múltiplos limiares ($K_1, K_2, K_3, \dots, K_n$) que irão pertencer a diferentes regiões ($R_1, R_2, R_3, \dots, R_{n+1}$) e realizam-se comparações para ver se o pixel em questão está dentro do limiar, por exemplo, dado um pixel $P(i,j)$, temos:

- Se $P(i,j) > 0$ & $P(i,j) \leq K_1$, então $P(i,j)$ pertence à R_1 ;
- Se $P(i,j) > K_1$ & $P(i,j) \leq K_2$, então $P(i,j)$ pertence à R_2 ;
- Se $P(i,j) > K_2$ & $P(i,j) \leq K_3$, então $P(i,j)$ pertence à R_3 ;
- Se $P(i,j) > K_n$, então $P(i,j)$ pertence à R_{n+1} ;

O método de clusterização, também conhecido como *k-means*, é um método de quantização vetorial que tem origem no processamento de sinais e é muito popular na análise de *clusters* na mineração de dados. O algoritmo consiste em separar os dados obtidos em K *clusters*. O número K equivale à quantidade de centroides que serão criados para cada grupo. Após isso computa-se o centro de cada *cluster* e a distância de cada ponto à este centro. Cada ponto deve pertencer ao cluster que tenha menor distância do seu centro. O algoritmo pode ser melhorado se assumirmos que os pontos vizinhos ao ponto analisado têm uma probabilidade alta de pertencerem ao mesmo cluster.



(a) Imagem aleatória



(b) Imagem agrupada

Figura 2.16: Exemplo do algoritmo *k-means* em cima de um conjunto de dados aleatório

Como podemos observar na Figura 2.16b, os dados foram agrupados em volta do centroide que possuía menor distância. É fácil de verificar que os dados gerados na

Figura 2.16a já possuem um agrupamento inicial, e fica fácil, neste caso, de determinar onde ficarão os centroides.

Uma outra maneira de se segmentar uma imagem pode ser através da análise do seu histograma. Este método é muito eficiente pois geralmente requer apenas uma iteração sobre os pixels da imagem. Nesta técnica, computa-se o histograma dos pixels da imagem e utiliza-se os picos e vales para a formação de *clusters*. Geralmente utilizam-se como parâmetros a cor ou a intensidade. Uma forma mais aprofundada desta técnica é aplicá-la recursivamente dentro dos *clusters* que vão se formando, até não existir mais nenhum. Uma desvantagem deste método é que pode ser difícil de se encontrar picos e vales em determinadas regiões de uma imagem.

Capítulo 3

Imagem por ressonância magnética

Ressonância magnética é uma técnica de obtenção de imagem muito utilizada na radiologia. Ela utiliza ondas magnéticas e de rádio para formar imagens de anatomia e processos fisiológicos do corpo humano, sendo amplamente utilizada em hospitais e clínicas, para diagnósticos médicos e acompanhamento de doenças [4].

Os pacientes que passam por este tipo de exame devem ter um acompanhamento, pois o exame não é indicado para gravidez recente e nem para pacientes que possuem alguns tipos de implantes (dispositivos que podem conter componentes elétrico ou magnéticos que não operam na faixa segura do aparelho de ressonância).

A ressonância magnética parte do princípio de que alguns núcleos atômicos podem absorver e emitir energia quando provocados por um campo magnético externo. Assim, o exame mapeia as moléculas de água e gordura no corpo, facilitando a formação de uma imagem rica em detalhes. Os dados obtidos podem possuir uma variedade de dados químicos e físicos, além de imagens detalhadas [9].

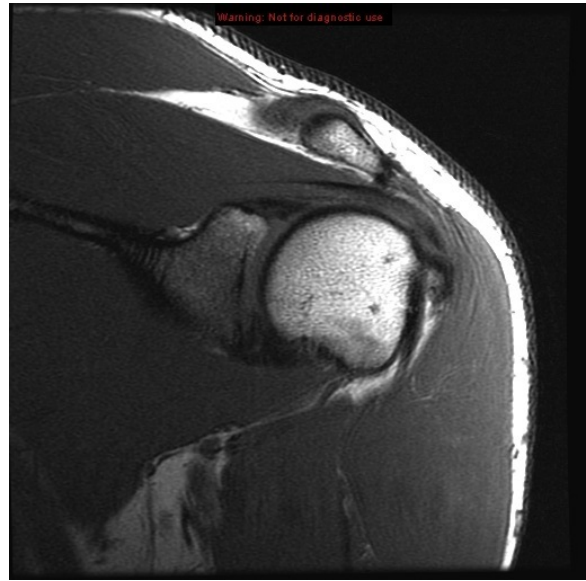
Essa técnica possui um número imenso de aplicações na área de diagnósticos médicos e é amplamente utilizada pois o paciente não é afetado por radiação ionizante. As áreas em que a ressonância é mais utilizada são:

- Imagens neurais;
- Imagens cardiovasculares;
- Imagens musculoesqueléticas;
- Imagens do fígado e do sistema gástrico;
- Imagens oncológicas.

Neste trabalho iremos focar em imagens do sistema musculoesquelético, mais especificamente imagens abdominais.



(a) Imagem do joelho



(b) Imagem do ombro

Figura 3.1: Exemplos de imagens por ressonância magnética[20]

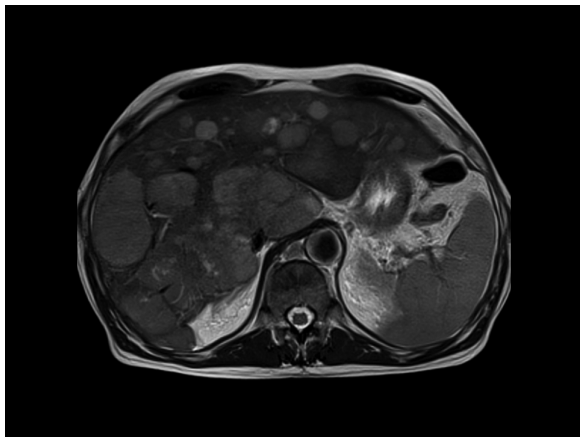
3.1 Imagens abdominais

A ressonância magnética abdominal é geralmente feita através de cortes transversais de 2 à 5 milímetros. Esses cortes podem ser utilizados para medição e identificação de tumores. Este exame pode ser utilizado para diversos tipos de diagnósticos, como:

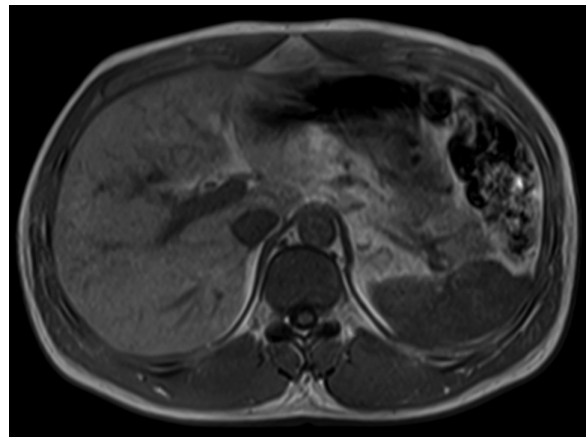
- Encontrar tumores nos órgãos ou tecidos que se encontram na região abdominal;
- Checar por tumores ou sangramentos na região pélvica;
- Checar órgãos e veias antes de um transplante;
- Encontrar bloqueios ou pedras nos tubos que levam bile do fígado para a vesícula biliar;
- Identificar cálculos renais e problemas de obstrução renal.

O exame pode ser realizado normalmente ou com a utilização de um contraste que tem aplicação intravenosa. O contraste age nas regiões que possuem vascularização, deixando estas regiões mais claras. Isso permite identificar mais facilmente de onde e para onde o fluxo sanguíneo de um tumor está direcionado.

A imagem gerada pela ressonância magnética também pode contrastar diferentes tipos de tecido do corpo humano. Uma imagem com peso T1 tem maior contraste nos tecidos moles, como órgãos: córtex cerebral, tecido gorduroso, fígado, rins. Imagens com peso T2 tendem a ter maior foco na matéria branca, ou seja, ossos e partes duras.



(a)



(b)

Figura 3.2: Exemplos de imagens do abdômen por ressonância magnética[19]

Capítulo 4

Revisão da literatura

A interpolação de imagens digitais é uma técnica amplamente utilizada e muito difundida em diferentes tecnologias nos dias de hoje. Podemos observar o seu uso em computadores, câmeras digitais, técnicas de processamento de imagem e, mais importante neste contexto, no processamento de imagens médicas. Existem dois métodos de interpolação que podem ser trabalhados, a adaptativa e a não-adaptativa. A interpolação adaptativa muda dependendo da área que estiver sendo interpolada (bordas, cantos, regiões contínuas), enquanto a não-adaptativa é aplicada igualmente sobre todas as regiões. Neste contexto, é fácil observar que a a interpolação adaptativa, por ser uma técnica muito mais robusta e mais complexa, é mais difícil de ser encontrada, principalmente em aplicações de uso comum. Por outro lado, é muito mais empregado na literatura técnicas de interpolação não-adaptativa, tanto pela facilidade de desenvolvimento como pelo desempenho oferecido por esta técnica.

Aplicações fora da área do uso comum, ou seja, aplicações técnicas e profissionais, podem precisar de qualidade extra e também podem dispor de um tempo de processamento mais elevado. Geralmente essas aplicações são softwares privados desenvolvidos para uso empresarial e não pessoal. Nesse tipo de aplicação podemos ver o uso de interpolação de imagens adaptativas. Ainda assim, seu ganho de qualidade não é tão grande para pagar o seu aumento de processamento, ainda sendo assim mais utilizada a não-adaptativa.

No meio médico, podemos observar claramente a utilização da interpolação adaptativa, pois a preocupação com qualidade é extrema. Muitas vezes, por imagens desse tipo terem um tamanho elevado e uma resolução muito acima das imagens comuns, o tempo de processamento é muito elevado. Em alguns casos, a urgência de um diagnóstico pode necessitar da priorização da performance sobre a qualidade.

Tratar esses extremos de qualidade e desempenho é uma tarefa muito complicada. O ideal seria ter uma aplicação para cada tipo de situação, porém, isso é impossível. Encontrar a dose certa entre esses parâmetros pode ser um cenário ideal, visto que pode-se obter imagens com qualidade elevada a um tempo de processamento não tão grande. Este tipo de aplicação pode ser tanto usado na área médica como em aplicações comuns. Hoje existem vários algoritmos que tentam realizar a junção de interpolação adaptativa e não adaptativa para realizar tal tarefa.

O *New Edge-Directed Interpolation* [13], **NEDI**, por exemplo, utiliza a interpolação bilinear e uma técnica adaptativa baseada na covariância. O algoritmo consegue alcançar

resultados visualmente e numericamente satisfatórios, porém, o tempo de processamento é elevado e a imagem retém artefatos indesejados em regiões de de frequências elevadas.

O *Edge-Guided Image Interpolation Algorithm via Directional Filtering and Data Fusion* [27], **EGII**, propõe uma interpolação não-linear focada nas bordas através de filtros direcionais e cruzamento de dados. Basicamente, o algoritmo determina que para um pixel ser interpolado, ele precisa de duas estimativas que possuam direções ortogonais diferentes. Essas estimativas são unidas utilizando a técnica *linear minimum square-error estimation* (LMMSE). O algoritmo também propõe uma versão com menor gasto computacional, substituindo o LMMSE por uma versão simplificada deste.

O *Iterative Curve Based Interpolation* [8], *Iterative Curve Based Interpolation* (**ICBI**), tem um foco maior em performance do que em qualidade. Este método descreve um novo jeito de se ampliar uma imagem baseado em um preenchimento da imagem em duas etapas e uma correção iterativa dos pixels obtidos através da interpolação. Esta interpolação é obtida através da função de derivadas direcionais de segunda ordem da imagem.

4.1 **TEDI**

O *True edge-directed interpolation* [26] (**TEDI**) utiliza um método voltado para o direcionamento das bordas. Primeiramente a imagem é interpolada pelo método bilinear. Após isso, o método *Canny* é utilizado para extrair as bordas e computar a direção destas. Comparando a imagem interpolada e a imagem original, borra-se as regiões onde as bordas não coincidem. Esse método não é computacionalmente pesado e mantém as estruturas de bordas da imagem bem preservadas. A grande peculiaridade do **TEDI** está na análise das bordas da imagem. As bordas são classificadas em quatro categorias:

- Direita-horizontal (RH) 0° - 45°
- Esquerda-vertical (LV) 45° - 90°
- Direita-vertical (RV) 90° - 135°
- Esquerda-horizontal (LH) 135° - 180°

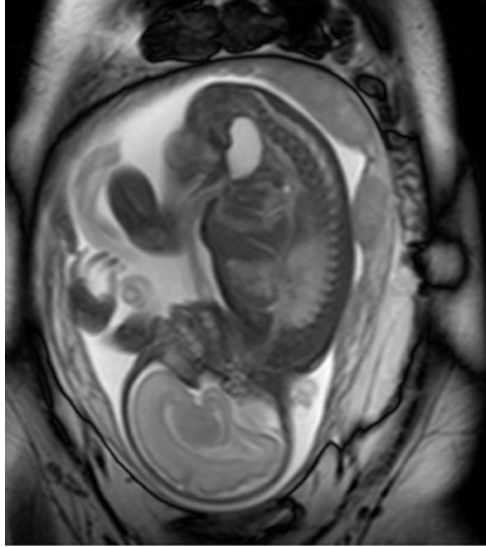


Figura 4.1: Imagem original



Figura 4.2: Mapa de bordas

Para obter o mapa de bordas 4.2 da imagem original 4.1, utiliza-se o Canny e depois separa-se essas bordas nas 4 diferentes orientações utilizadas pelo **TEDI**.

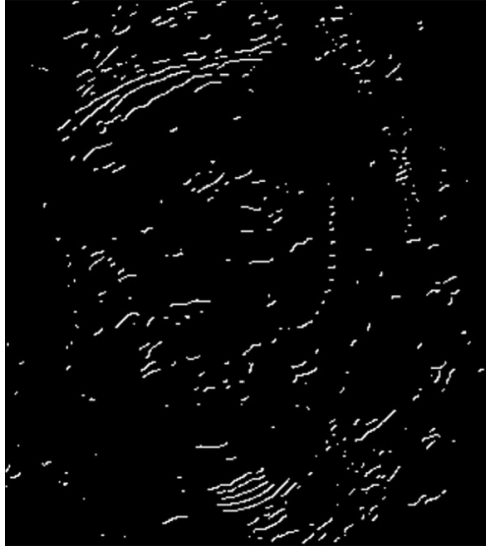


Figura 4.3: Bordas 0° - 45° (RH)

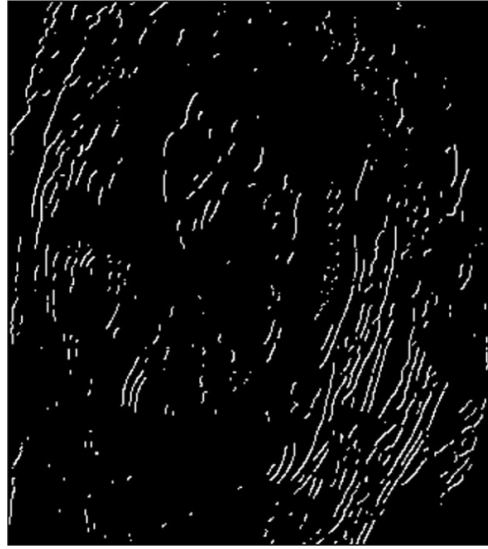


Figura 4.4: Bordas 45° - 90° (LV)



Figura 4.5: Bordas 90° - 135° (RV)



Figura 4.6: Bordas 135° - 180° (LH)

Após a obtenção das bordas com direcionamento, compara-se elas com as bordas da imagem original interpolada. Essa comparação pode resultar em:

- A borda existe em ambas as imagens. Neste caso, aplica-se um filtro direcional 5x5, como mostra a Figura 4.7.
- A borda existe em uma imagem e não existe na outra. Neste caso, aplica-se um filtro de média para suavizar a região.

1	2	1	0	0
0	1	1	0	0
0	0	4	0	0
0	0	1	1	0
0	0	1	2	1

0° - 45° (RH)

1	0	0	0	0
2	1	0	0	0
1	1	4	1	1
0	0	0	1	2
0	0	0	0	1

45° - 90° (LV)

0	0	0	0	1
0	0	0	1	2
1	1	4	1	1
2	1	0	0	0
1	0	0	0	0

90° - 135° (RV)

0	0	1	2	1
0	0	1	1	0
0	0	4	0	0
0	1	1	0	0
1	2	1	0	0

135° - 180° (LH)

Figura 4.7: Máscaras aplicadas sobre os pontos analisados nas Figuras 4.3 4.4 4.5 4.6

O **TEDI** apresenta um valor de qualidade muito alto quando comparado aos outros algoritmos apresentados.

4.2 LAI

O *Locally Adaptive Non-Linear Interpolation*, **LAI** [16], é um algoritmo que prioriza muito a rapidez e simplicidade de execução, mantendo um padrão de qualidade suficiente e que sobrepõe diversos métodos na literatura. O **LAI** é executado em quatro etapas, sendo elas, expansão (1), preservação de bordas direcional (2), alisamento por gradiente (3) e preenchimento por gradiente (4).

1. Expansão

Nesta fase a imagem original **I** é simplesmente expandida para o dobro do seu tamanho menos um. As coordenadas ímpares da nova imagem **Z** correspondem

as coordenadas da imagem original. Ou seja, $Z(2x,2y) = I(x,y)$, como podemos observar na figura 4.8.

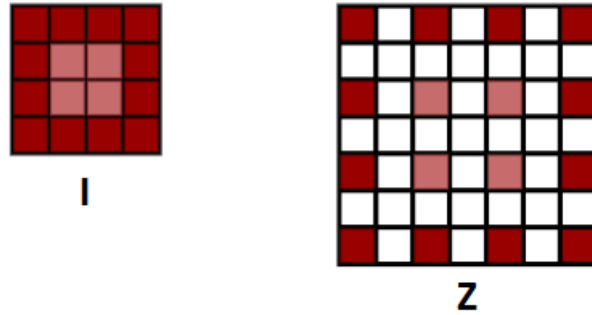


Figura 4.8: Fase de expansão do LAI

2. Preservação de bordas direcional

O objetivo desta fase é identificar bordas na imagem original baseado na vizinhança próxima de cada pixel de intersecção central. Para isso, são considerados cinco casos de angulação das bordas:

- Borda horizontal (0°);
- Borda vertical (90°);
- Borda esquerda-diagonal (135°);
- Borda direita-diagonal (45°);
- Sem borda.

Na figura 4.9 podemos observar a nomenclatura dada para cada ponto após o processo de expansão da imagem original. A partir dos valores desses pixels, calcula-se dois valores de referência T1 e T2 que ajudam na identificação do tipo de borda.

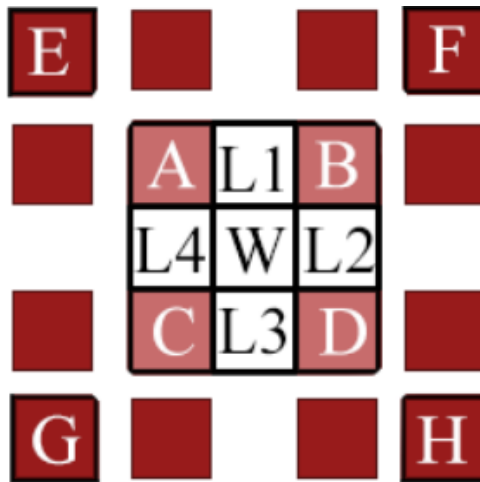


Figura 4.9: Nomenclatura dos pontos da imagem expandida
Imagem retirada do texto do algoritmo LATEDI [15]

Os valores de T1 e T2 são calculados da seguinte maneira:

$$T1 = std(A, B, C, D, E, F, G, H) \quad (4.1)$$

$$T2 = \frac{T1}{\sqrt{2}} \quad (4.2)$$

Onde **std** representa o desvio padrão dos valores. A partir desses valores, podemos calcular o valor dos pixels em W (coordenadas pares de Z):

- (a) Se $|amplitude(A, B, C, D)| < T1$,
então $W = \frac{A+B+C+D}{4}$
- (b) Se $|A-D| > T2 \ \& \ |A-D| \gg |B-C|$,
então $W = \frac{B+C}{2}$
- (c) Se $|B-C| > T2 \ \& \ |B-C| \gg |A-D|$,
então $W = \frac{A+D}{2}$
- (d) Se $|A-D| > T1 \ \& \ |B-C| > T1 \ \& \ (A-D)*(B-C) > 0$,
então $L1 = \frac{A+B}{2}; L3 = \frac{C+D}{2}$
- (e) Se $|A-D| > T1 \ \& \ |B-C| > T1 \ \& \ (A-D)*(B-C) < 0$,
então $L4 = \frac{A+C}{2}; L2 = \frac{B+D}{2}$

3. Alisamento por gradiente

Na fase anterior, os pontos que possuíam ao menos uma coordenada ímpar não foram preenchidos. Nesta fase, calculamos os mesmos valores de T1 e T2 como na anterior porém para cada pixel Q (pontos da imagem **Z** com ao menos uma coordenada ímpar). Consideraremos a vizinhança como na Figura 4.10



Figura 4.10: Vizinhança dos pontos com ao menos uma coordenada ímpar
Imagem retirada do texto do algoritmo **LATEDI** [15]

Os casos abordados nesta etapa são:

Primeiro caso: L1 e L3 são conhecidos:

- (a) Se L2 ou L4 estiverem em branco
Se $|L1-L3| < T1$,
então $Q = \frac{L1+L3}{2}$

- (b) Se L2 ou L4 são conhecidos
 1. Se $|L1-L3| > T2$ & $|L1-L3| \gg |L2-L4|$,
então $Q = \frac{L2+L4}{2}$
 2. Se $|L2-L4| > T2$ & $|L2-L4| \gg |L1-L3|$,
então $Q = \frac{L1+L3}{2}$

Segundo caso: L2 e L4 são conhecidos:

- (a) Se L1 ou L3 estiverem em branco

Se $|L2-L4| < T1$,
então $Q = \frac{L2+L4}{2}$
- (b) Se L1 ou L3 são conhecidos
 1. Se $|L2-L4| > T2$ & $|L2-L4| \gg |L1-L3|$,
então $Q = \frac{L1+L3}{2}$
 2. Se $|L1-L3| > T2$ & $|L1-L3| \gg |L2-L4|$,
então $Q = \frac{L2+L4}{2}$

Ainda assim, pode acontecer de nenhuma dessas condições serem satisfeitas e o ponto **Q** continuar em branco.

4. Preenchimento por gradiente Para finalizar o preenchimento dos pontos que ainda ficaram em branco, fazemos o seguinte:
 - (a) Se o ponto possui ambas coordenadas pares,
então W = média aritmética das medianas das janelas dos pontos diagonais A,B,C e D
 - (b) Se o ponto possui ao menos uma coordenada ímpar,
então Q = média aritmética das medianas das janelas dos pontos horizontais e verticais L1, L2, L3 e L4.

Após realizar todas estas etapas, caso ainda exista algum ponto não preenchido, o algoritmo é executado novamente.

4.3 LATEDI

O Locally Adaptive True Edge-Directed Interpolation, **LATEDI** [15], foi criado a partir de uma junção do **LAI** e do **TEDI** com algumas pequenas modificações. A primeira fase do algoritmo é executar uma versão levemente modificada do **LAI**. Na parte de expansão do algoritmo, aumenta-se a imagem **I** para o dobro do número de linhas e colunas, ao invés do dobro menos um como proposto pelo **LAI**.

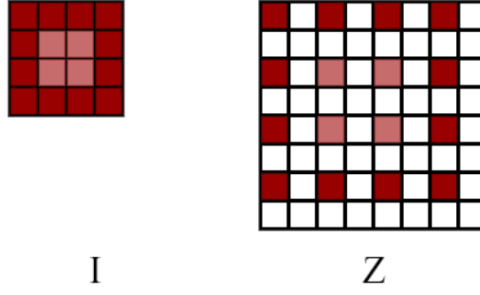


Figura 4.11: Fase de expansão do **LATEDI**
 Imagem retirada do texto do algoritmo **LATEDI** [15]

Outra mudança realizada na forma como o **LAI** é executado está no cálculo dos parâmetros T1 e T2. Usando como referência a Figura 4.9, o **LAI** é calculado com o desvio padrão dos valores (A, B, C, D, E, F, G, H). Nesta nova abordagem, o cálculo fica da seguinte forma:

$$T1 = \frac{amplitude(A, B, C, D, E, F, G, H)}{3} \quad (4.3)$$

$$T2 = \frac{T1}{\sqrt{2}} \quad (4.4)$$

Depois desta etapa, o **LAI** segue sua execução normal utilizando os novos valores calculados de T1 e T2. Após a finalização do **LAI**, o algoritmo **TEDI**, que é mencionado no **Seção 4.1** é executado com uma leve modificação: a matriz de borda via detector Canny é calculada apenas para a imagem duplicada Z, e não há subtração das bordas originais. Assim temos o resultado final do **LATEDI**.

Capítulo 5

Metodologia

A partir da análise dos algoritmos citados nos capítulos anteriores, podemos ver que o TEDI apresenta uma melhor qualidade de imagem. A ideia é melhorar o resultado do TEDI mesclando as técnicas apresentadas. Serão apresentadas duas vertentes do TEDI. Ambas misturam o conceito de segmentação com o algoritmo. A primeira, batizada de $TEDI_s$, divide a imagem em diferentes regiões, a partir da segmentação, depois aplica-se o canny na imagem segmentada e a partir deste resultado o algoritmo TEDI é passado na imagem. Já o $TEDI_a$, parte do mesmo princípio de primeiramente realizar a segmentação da imagem e aplicar o canny. Porém, após este processo, aplica-se uma mistura dos algoritmos TEDI e LAI, que será detalhado em seguida.

A segmentação aplicada divide a imagem **I** em 4 regiões diferentes, resultando na imagem **Z**:

- se $I(x, y) \geq 0$ & $Z(x, y) < 100$;
 $X(x, y) = 0$
- se $I(x, y) \geq 100$ & $Z(x, y) < 180$;
 $X(x, y) = 120$
- se $I(x, y) \geq 180$ & $Z(x, y) < 220$;
 $X(x, y) = 210$
- se $I(x, y) \geq 220$ & $Z(x, y) \leq 255$;
 $X(x, y) = 255$

Para o conjunto de imagens utilizado, as regiões ficaram bem definidas com esta segmentação. A imagem é separada em partes como osso, órgãos e regiões internas, como podemos observar na figura 5.2. Os níveis escolhidos não são arbitrários, eles são tons que representam partes que se destacam na imagem.



Figura 5.1: Imagem original

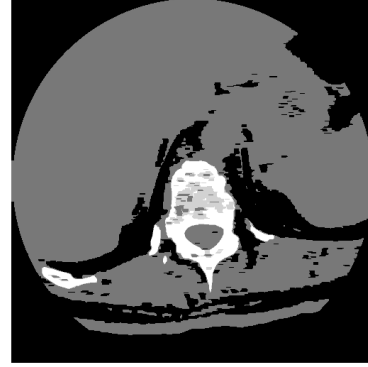


Figura 5.2: Imagem segmentada



Figura 5.3: Imagem original



Figura 5.4: Imagem segmentada

A etapa de segmentação é feita para se remover a grande quantidade de bordas que são formadas pelo ruído que se encontra na imagem original. Observando a Figura 5.3 podemos ver várias estruturas na imagem que não representam nada, ou seja, falsas bordas. Na Figura 5.4 as bordas são muito mais bem definidas e podem representar detalhes importantes da imagem como órgãos, ossos e até tumores.

Concluindo-se a segmentação, a próxima etapa é realizar a detecção de bordas direcionais, como explicado na Seção 4.1. Até esta etapa ambos o $TEDI_s$ e o $TEDI_a$ são idênticos. As próximas etapas apresentam diversas mudanças no algoritmo.

5.1 $TEDI_s$

O $TEDI_s$ passa por um pré-processamento, que resulta em duas imagens. A primeira imagem (**Bordas1**) é fruto das bordas da segmentação da imagem original. A segunda imagem (**Bordas2**) representa as bordas da imagem original interpolada. Pode-se verificar este pré-processamento a partir do diagrama ilustrado na Figura 5.5

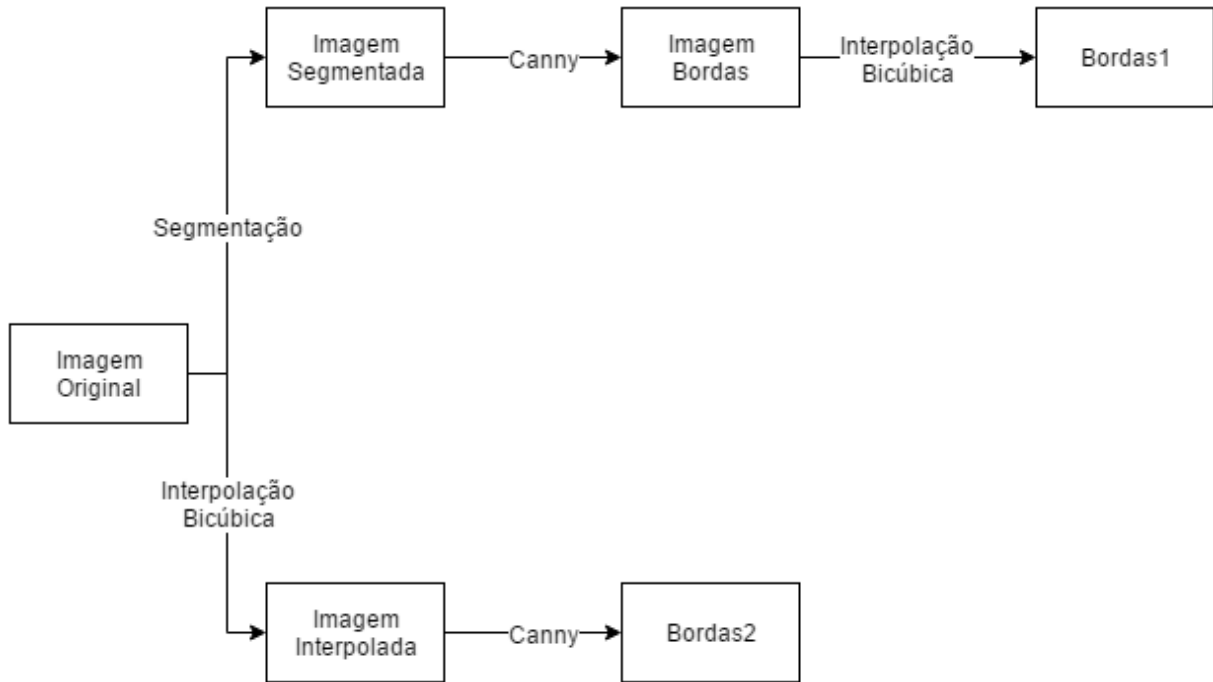


Figura 5.5: Primeira etapa do $TEDI_s$

A partir da imagem **Bordas2**, o algoritmo verifica a qual direção pertence a borda, como explicado na [Seção 4.1](#):

- Direita Horizontal (RH);
- Esquerda Vertical (LV);
- Direita Vertical (RV);
- Esquerda Horizontal (LH);

Depois é verificado se essa mesma borda existe na imagem **Bordas1**. Caso ela exista é aplicado um dos filtros da imagem [4.7](#), senão é aplicado um filtro de média sobre a vizinhança 5x5, para suavizar a borda.

5.2 $TEDI_a$

O método do $TEDI_s$ é muito parecido com o TEDI, porém ele apresenta algumas diferenças, como a comparação com a imagem segmentada. O $TEDI_a$ já mostra diversas diferenças, até mesmo nos filtros que são passados no caso de detecção de borda.

Inicialmente o algoritmo passa pela mesma fase inicial do $TEDI_s$, porém, a imagem **Bordas2** é gerada de uma forma levemente diferente, como pode ser visto na figura [5.6](#). Antes de ela ser interpolada pelo método bicúbico, a imagem passa por um detector de bordas Canny.

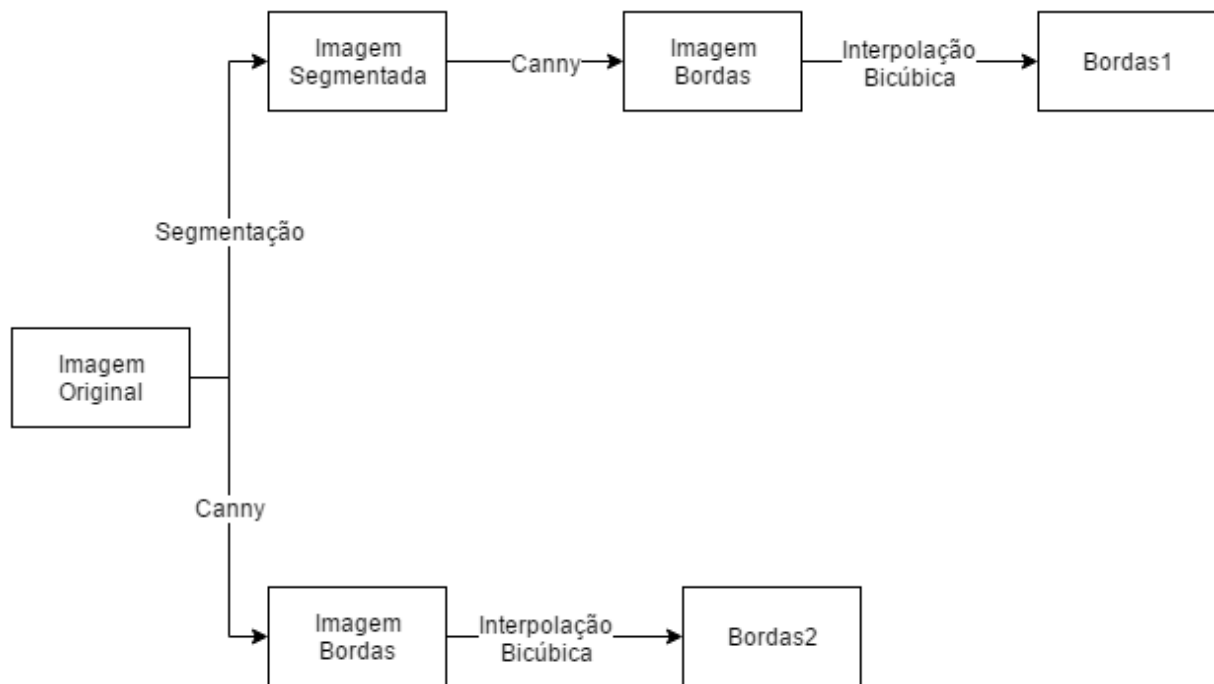


Figura 5.6: Primeira etapa do $TEDI_s$

Após esta etapa, o algoritmo verifica se para cada pixel em **Bordas2** existe o mesmo pixel em **Bordas1**, e isso nos leva à dois casos:

- Caso o pixel não exista em ambas as imagens, é aplicado um filtro de média sobre a vizinhança 5x5 do valor em questão.
- Caso o pixel exista em ambas as imagens, é aplicado um filtro muito semelhante ao da primeira parte do LAI, que está representado na Figura 5.7.

E				F
	A	L1	B	
	L4	W	L2	
	C	L3	D	
G				H

Figura 5.7: Vizinhança 5x5 considerada

O filtro foi originalmente mudado do tamanho 7x7 para 5x5. Isso oferece uma razão entre qualidade e desempenho melhor do que o proposto pelo **LAI**. Para calcular os valores de **T1** e **T2**, utilizamos:

$$T1 = std(A, B, C, D, E, F, G, H) \quad (5.1)$$

Onde *std* representa o desvio padrão dos valores, e:

$$T2 = \frac{T1}{\sqrt{2}} \quad (5.2)$$

A partir desses valores podemos encontrar o valor de W para cada caso:

1. Se $|amplitude(A,B,C,D)| < T1$,
então $W = \frac{A+B+C+D}{4}$
2. Se $|A-D| > T2$ & $|A-D| \gg |B-C|$,
então $W = \frac{B+C}{2}$
3. Se $|B-C| > T2$ & $|B-C| \gg |A-D|$,
então $W = \frac{A+D}{2}$

4. Se $|A-D| > T1$ & $|B-C| > T1$ & $(A-D)*(B-C) > 0$,
então $L1 = \frac{A+B}{2}$; $L3 = \frac{C+D}{2}$
5. Se $|A-D| > T1$ & $|B-C| > T1$ & $(A-D)*(B-C) < 0$,
então $L4 = \frac{A+C}{2}$; $L2 = \frac{B+D}{2}$

Capítulo 6

Resultados

Para realizar a comparação dos resultados, os algoritmos **TEDI**, **NEDI**, **EGII** e **LAI** citados no **Capítulo 4** também foram executados e testados. Também foram utilizados algoritmos de métricas para obter certos parâmetros de comparação:

- *Peak signal-to-noise ratio* (**PSNR**) é a razão entre a potência máxima de um sinal e a potência do ruído deste. Geralmente é expresso em decibel;
- *Structural similarity index* [23] (**SSIM**) é um modelo baseado na percepção que considera a degradação da imagem como uma mudança percebida na informação estrutural;
- *Feature similarity index* [28] (**FSIM**) utiliza a análise da congruência de fase para analisar a qualidade da imagem, desta forma, o algoritmo aproxima-se mais da percepção visual humana.

As métricas são aplicadas utilizando-se a imagem original e a imagem final gerada por cada algoritmo. Os códigos gerados foram testados na plataforma MATLAB R2016a, em um PC com sistema operacional Windows 10, processador Intel Core i7-4820K @ 3.70GHz, com 16GB de Memória RAM.

As imagens de teste foram fornecidas pelo Hospital Universitário de Brasília, e os pacientes representados nas imagens não são identificados. Foi utilizado um conjunto de 20 cortes transversais de imagens de ressonância magnética que estavam no formato DICOM (*Digital Imaging and Communications in Medicine*). Este formato é utilizado para manipular, armazenar, guardar, imprimir e transmitir imagens médicas. Ele inclui um formato de arquivo e um protocolo de comunicação de rede[2].



Figura 6.1: Imagem original



Figura 6.2: **TEDI**



Figura 6.3: **LAI**



Figura 6.4: **NEDI**



Figura 6.5: **EGII**



Figura 6.6: $TEDI_a$



Figura 6.7: $TEDI_s$

Em uma primeira análise visual, é difícil enumerar as diferenças entre as imagens. Porém, se olharmos com atenção podemos observar uma leve diferença na parte branca central da imagem. Os algoritmos **TEDI**, $TEDI_s$ e $TEDI_a$ mantêm uma fidelidade maior à imagem original, enquanto os outros acabam realizando um borrimento maior nesta parte. Por focarem mais na suavização do que em manter as bordas, os detalhes acabam ficando distorcidos, prejudicando a composição original da imagem.

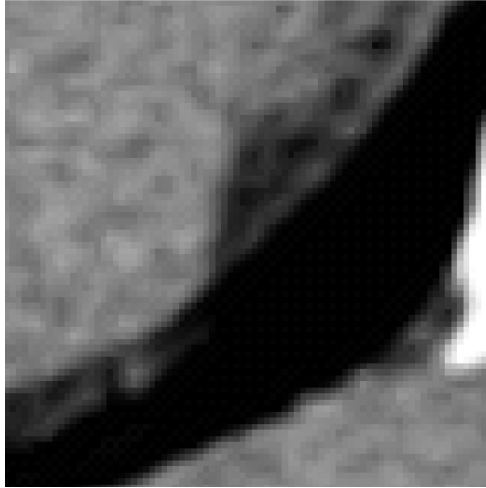


Figura 6.8: **LAI**

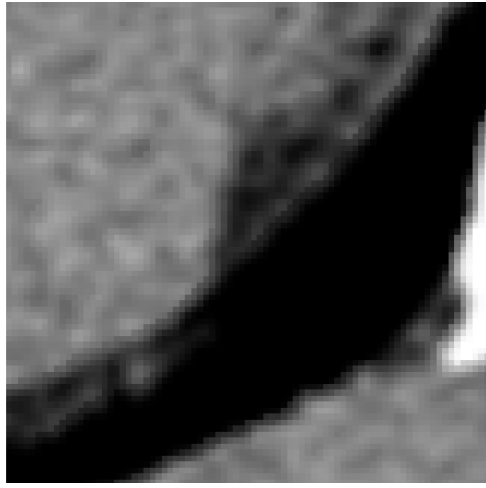


Figura 6.9: **TEDI**

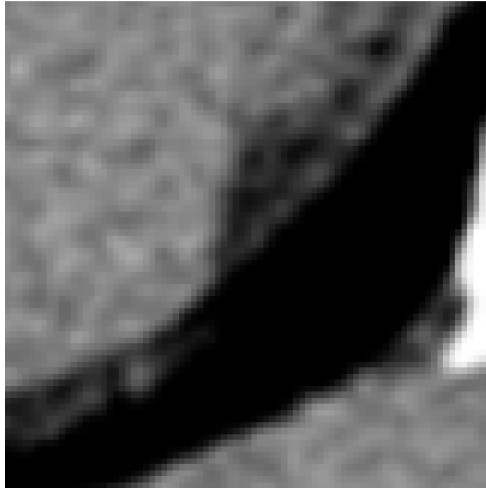


Figura 6.10: $TEDI_a$

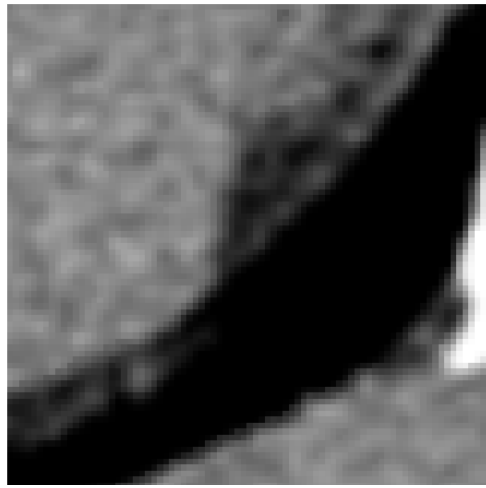
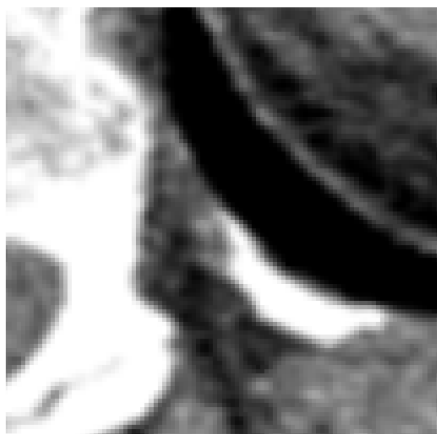


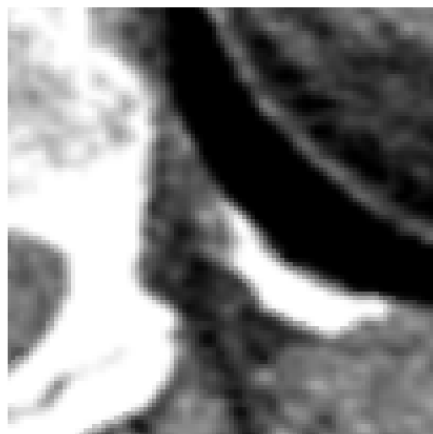
Figura 6.11: $TEDI_s$

Ao se comparar o **LAI** com os derivados do **TEDI**, podemos observar que as estruturas da imagem não são bem definidas. Isso acontece graças a maior preservação e importância que o **TEDI** dá as bordas. Podemos observar que o **TEDI** possui um leve sombreamento nas regiões das bordas. Essa característica é reduzida no $TEDI_s$ e praticamente eliminada no $TEDI_a$. As bordas mais bem definidas do $TEDI_s$ são muito úteis quando se está fazendo a medição ou acompanhamento de um tumor, por exemplo. No caso do $TEDI_a$, as bordas mais alisadas podem oferecer maior segurança em um diagnóstico, pois estruturas muito pequenas (que se formam graças ao ruído da imagem), não ficam tão evidentes, não formando estruturas que não existam na imagem.

Ainda em uma análise visual, podemos ver que o $TEDI_s$ e o $TEDI_a$ preservam melhor as bordas originais da imagem do que os outros algoritmos. O $TEDI_a$ providencia um conforto visual melhor, visto que as bordas pixeladas são eliminadas devido à um maior borramento local feito na imagem. Já por outro lado, o $TEDI_s$ apresenta uma maior fidelidade à imagem original, justamente por não realizar um alisamento tão grande.



(a) $TEDI_a$



(b) $TEDI_s$

Figura 6.12: Comparação entre as bordas das imagens geradas pelos respectivos algoritmos

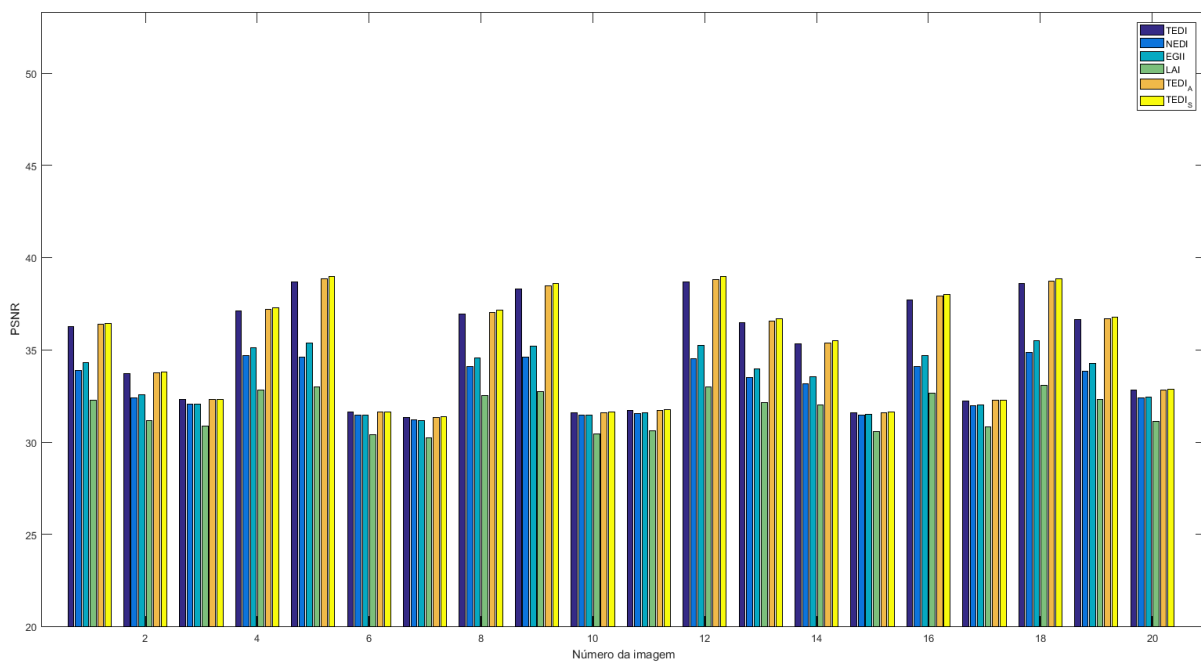


Figura 6.13: PSNR

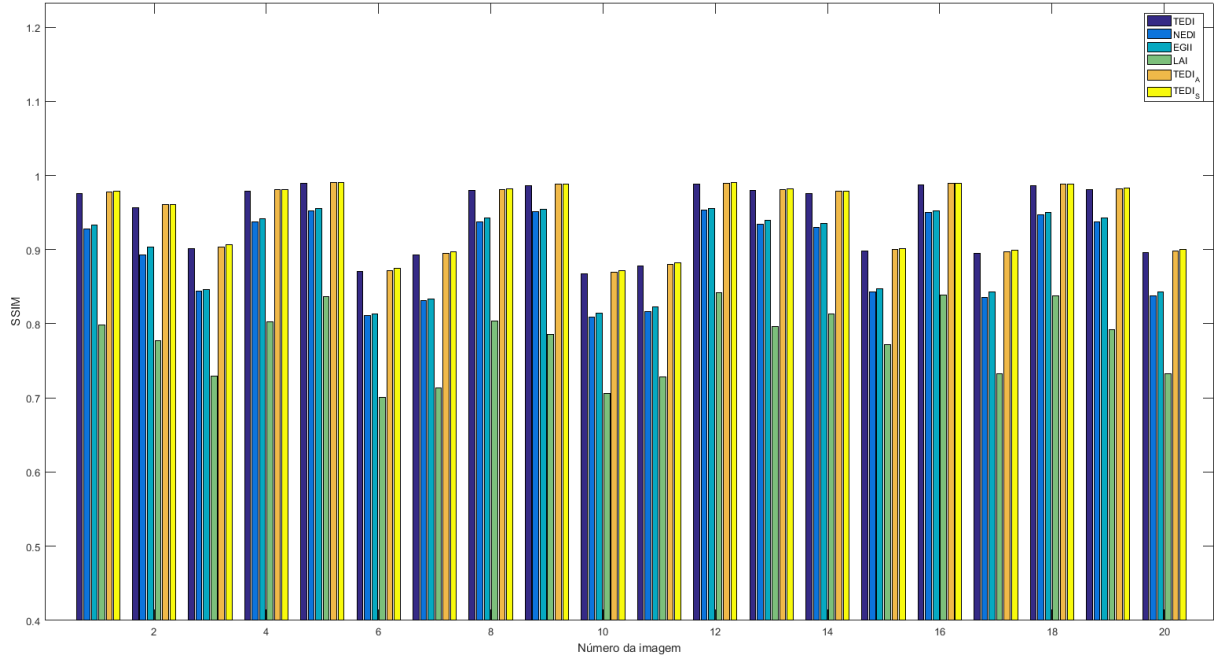


Figura 6.14: **SSIM**

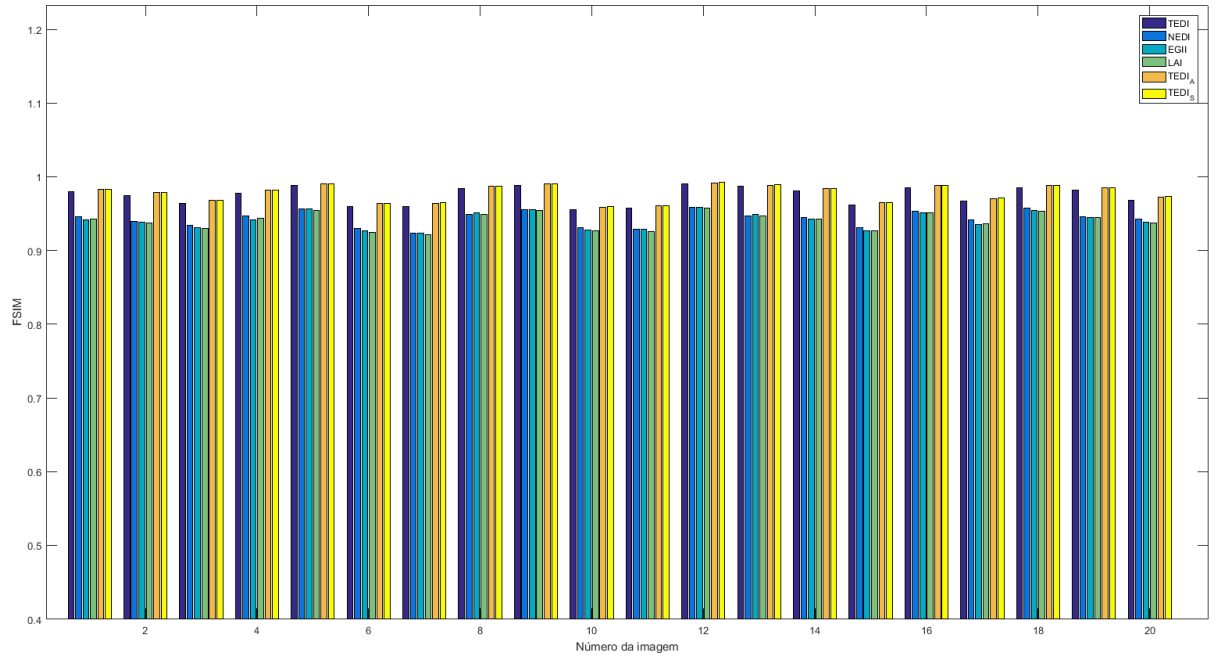


Figura 6.15: **FSIM**

Partindo para uma análise numérica, o $TEDI_s$ se destaca. Ele mantém a qualidade do algoritmo **TEDI** e fica ainda mais eficiente quando misturado com a segmentação inicial aplicada. Visto que o $TEDI_a$ é um algoritmo muito diferente do **TEDI**, e que o **LAI** (que possui desempenho numérico muito inferior) é uma das bases deste algoritmo, podemos concluir que o ganho deste é notável. Por realizar um borramento maior nas bordas da imagem, ele acaba por possuir métricas inferiores ao $TEDI_s$.

Tabela 6.1: Média dos valores obtidos com cada algoritmo

	NEDI	LAI	EGII
PSNR	33.0906 ± 1.3140	31.7376 ± 1.0301	33.4013 ± 1.5793
SSIM	0.8938 ± 0.0569	0.7768 ± 0.0474	0.8984 ± 0.0568
FSIM	0.9432 ± 0.0105	0.9404 ± 0.0114	0.9411 ± 0.0112
	TEDI	TEDIs	TEDIa
PSNR	34.9837 ± 2.8364	35.1278 ± 2.9454	35.0576 ± 2.8998
SSIM	0.9432 ± 0.0481	0.9462 ± 0.0470	0.9452 ± 0.0479
FSIM	0.9749 ± 0.0120	0.9782 ± 0.0113	0.9780 ± 0.0113

Analisando o tempo de execução médio dos algoritmos, podemos ver claramente que as alterações feitas no algoritmo **TEDI** não influenciam muito o seu desempenho. A tabela 6.2 mostra essa diferença. Como pode-se observar, todos os algoritmos derivados do **TEDI** possuem um tempo de execução muito menor, sendo uma vantagem crucial para algumas aplicações. A segmentação inicial aplicada resulta em uma diferença de tempo em ambos $TEDI_a$ e $TEDI_s$. O $TEDI_a$ apresenta um maior tempo de processamento visto que ele possui um maior número de comparações na aplicação do filtro 5x5.

Tabela 6.2: Tempo de execução médio dos algoritmos

	NEDI	LAI	EGII	TEDI	TEDIs	TEDIa
Tempo(s)	12.2026	1.2813	8.2942	0.2447	0.2623	0.3682

Capítulo 7

Conclusões

O desempenho dos algoritmos $TEDI_a$ e $TEDI_s$ são satisfatórios, visto que eles ultrapassam o famoso TEDI na avaliação numérica, e realiza algumas mudanças visuais nas imagens, como diminuição dos serrilhados nas bordas. O foco em preservação de bordas é mantido e o tempo de execução do algoritmo é levemente maior.

Visto que o algoritmo **TEDI** é muito eficaz e muito conhecido na literatura pela sua qualidade, obter métricas melhores do que ele pode ser considerado um avanço interessante. Vale notar também que, no caso do $TEDI_a$, as bordas são mais suavizadas, o que pode ser benéfico em aplicações médicas diferentes.

Portanto, seria interessante que o ganho nas métricas fosse superior, visto que o algoritmo passa por uma segmentação prévia e tem o seu tempo de execução aumentado. No caso do $TEDI_s$ a perda é mínima, pois a estrutura do algoritmo foi levemente modificada, porém no $TEDI_a$ podemos notar uma diferença por volta de 0.1 segundos à mais no tempo de processamento.

No final, pode-se ver que o ganho visual se tornou mais interessante do que o ganho nas métricas, pois foram apresentados situações em que cada imagem se beneficiaria em uma análise médica.

7.1 Trabalhos futuros

O objetivo é sempre aprimorar o algoritmo e têm-se como prioridade a melhoria na parte de segmentação da imagem e uma análise mais profunda nos filtros utilizados pelo $TEDI_a$. No $TEDI_s$ deve ser adicionado novas direções para os casos de detecção de borda. No quesito de tempo de processamento, seria ideal que o algoritmo fosse codificado em C++.

Referências

- [1] Tech Algorithm. <http://tech-algorithm.com/articles/nearest-neighbor-image-scaling/>. 8
- [2] National Electrical Manufacturers Association et al. *Digital imaging and communications in medicine (DICOM)*. NEMA, 2009. 32
- [3] Gabriel I Barbash and Sherry A Glied. New technology and health care costs—the case of robot-assisted surgery. *New England Journal of Medicine*, 363(8):701–704, 2010. 1
- [4] ALBS Bathista and JS Nogueira. Elementos históricos da ressonância magnética nuclear. *VII Jornada Brasileira de Ressonância Magnética Nuclear*, 2002. 14
- [5] Super Computing Blog. <http://supercomputingblog.com/graphics/coding-bilinear-interpolation/>. vi, 9
- [6] NDT Resource Center. History of radiography. <https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/Introduction/history.htm>. Accessed: 2017-01-27. 1
- [7] EMBS. www.embs.org/about-biomedical-engineering/our-areas-of-research/diagnostic-therapeutic-systems/. vi, 2
- [8] Andrea Giachetti and Nicola Asuni. Real-time artifact-free image upscaling. *IEEE Transactions on Image Processing*, 20(10):2760–2768, 2011. 18
- [9] VMS Gil and CFGC Geraldês. Ressonância magnética nuclear: Fundamentos, métodos e aplicações. 1987. *Lisboa: Fundação Calouste Gulbenkian*, 1012. 14
- [10] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. 2
- [11] Cambridge in Colour. <http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>. 6
- [12] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, Dec 1981. 3
- [13] Xin Li and M. T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10(10):1521–1527, Oct 2001. 17

- [14] John L Ostuni, Attanagoda KS Santha, Venkata S Mattay, Daniel R Weinberger, Ronald L Levin, and Joseph A Frank. Analysis of interpolation effects in the reslicing of functional mr images. *Journal of computer assisted tomography*, 21(5):803–810, 1997. 7
- [15] Guilherme F. Peres. Um algoritmo localmente adaptativo direcionado a bordas para interpolação de imagens digitais. 22, 23, 24, 25
- [16] L. Rodrigues, D. Leandro Borges, and L. Marcos Gonalves. A locally adaptive edge-preserving algorithm for image interpolation. In *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on*, pages 300–305, 2002. 21
- [17] S Schaller and T Flohr. Computed tomography—past, present and future. In *Multislice CT*, pages 3–17. Springer, 2004. 2
- [18] Linda Shapiro and George C Stockman. Computer vision. 2001. ed: Prentice Hall, 2001. 11
- [19] Siemens. <https://www.healthcare.siemens.com/magnetic-resonance-imaging/0-35-to-1-5t-mri-scanner/magnetom-essenza/use>. vi, 16
- [20] Stanford. <http://xrayhead.com/>. vi, 15
- [21] P. Thévenaz, T. Blu, and M. Unser. *Image Interpolation and Resampling*. Academic Press, San Diego CA, USA, 2000. 5
- [22] Philippe Thevenaz, Urs E Ruttimann, and Michael Unser. A pyramid approach to subpixel registration based on intensity. *Image Processing, IEEE Transactions on*, 7(1):27–41, 1998. 7
- [23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 32
- [24] Frederick M. Weinhaus and Venkat Devarajan. Texture mapping 3d models of real-world scenes. *ACM Comput. Surv.*, 29(4):325–365, December 1997. 7
- [25] Sholom M Weiss, Casimir A Kulikowski, Saul Amarel, and Aran Safir. A model-based method for computer-aided medical decision-making. *Artificial intelligence*, 11(1-2):145–172, 1978. 1
- [26] Wu S Hu J Xie Y Yu S, Zhang R. An edge-directed interpolation method for fetal spine mr images. *BioMedical Engineering OnLine*, 12:102, 2013. 3, 18
- [27] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE transactions on Image Processing*, 15(8):2226–2238, 2006. 3, 18
- [28] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: a feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 32

- [29] X. Zhang and X. Wu. Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation. *IEEE Transactions on Image Processing*, 17(6):887–896, June 2008. 3